FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 36, N° 2, June 2023, pp. 285-298
https://doi.org/10.2298/FUEE2302285R

Original scientific paper

A STUDY OF BINARY DECISION DIAGRAM CHARACTERISTICS OF BENT BOOLEAN FUNCTIONS

Miloš Radmanović

Faculty of Electrical Engineering, University of Niš, Niš, Serbia

Abstract. *Bent Boolean functions exist only for an even number of variables, moreover, they are unbalanced. Therefore, they are used in coding theory and in many areas of computer science. General form of bent functions is still unknown. One way of representing Boolean functions is with a reduced ordered binary decision diagram (ROBDD). The strength of ROBDDs is that they can represent Boolean functions data with a high level of redundancy in a compact form, as long as the data is encoded in such a way that the redundancy is exposed. This paper investigates characteristics of bent functions with focus on their ROBDD parameters. Decision diagram experimental framework has been used for implementation of a program for calculation of the ROBDD parameters. The results presented in this paper are intended to be used to create methods for the construction of bent functions using a ROBDD as a data structure from which the bent functions can be discovered.*

Key words: *Coding theory, Boolean functions, bent functions, binary decision diagram*

1. INTRODUCTION

Bent functions were introduced by Rothaus in 1976 as Boolean functions with maximal nonlinearity [1]. This feature has led to the fact that they are used in many areas of computer science, but their most significant application is in coding theory [2]. However, their application is quite limited because the general form of these functions is still unknown [3]. As the number of variables in a function increases, bent functions become extremely rare in the set of all possible functions. Therefore, the construction of these functions is computationally demanding. Also, there is no formal method for enumeration, generalization, construction, or classification of all bent functions for a given number of variables. For this reason, researching the characteristics of these functions is of great interest to the scientific community.

Some characteristics of bent functions are known. It is well known that all Walsh spectral coefficients of bent functions, whose values are mapped onto {-1, 1}, have the same absolute value $2^{n/2}$, where n is the number of variables of the function. They have the maximum possible value of nonlinearity equal to $(2^{n-1} \pm 2^{(n/2)-1})$, and they only exist

Received November 01, 2022; revised December 30, 2022 and March 14, 2023; accepted April 10, 2023
Corresponding author: Miloš Radmanović
Faculty of Electrical Engineering, University of Niš, Aleksandra Medvedeva 14, 18104 Niš, Serbia
E-mail: milos.radmanovic@elfak.ni.ac.rs

for an even number of variables. It is known that the algebraic degree of bent functions is at most $n/2$ for $n \geq 4$ [3]. It is also well known that from given bent functions, new bent functions with the same or a greater number of variables can be constructed [4]. Extensive work on bent functions has been done and various interesting results of research have been brought out in respect of construction and classification of specific subsets of bent functions [3], [5].

As the number of variables increases, it is very time consuming to discover a bent function. The number of bent functions is small compared to the complete set of all Boolean functions and therefore, their detection in some defined boundaries is very time-consuming. Testing of bentness across all possible functions, using all known characteristics, even for small numbers of variables ($n>4$), requires a lot of processing time.

Consequently, the complete set of n-variable bent Boolean functions is known only for $n \leq 4$ [3]. The general number of bent functions is an open problem. Note that the number of bent functions increases rapidly with increasing n. There are 8 bent functions in 2 variables, 896 bent functions in 4 variables, 5,425,430,528 bent functions in 6 variables, and 99,270,589,265,934,370,305,785,861,242,880 bent functions in 8 variables [5]. For example, for a complete enumeration of bent functions with 8 variables (which is approximately $8.57*10^{-44}$ percent of all functions), using maximal algebraic degree, has used approximately 50 personal computers running for 3 months [6]. For this reason, identifying new characteristics can help in more efficient detection of bent functions.

One way to represent a Boolean function is with a binary decision diagram (BDD). BDDs were originally introduced as an efficient support to the procedures and operations required to solve a given problem in synthesis and verification of logic circuits [7]. They are popular data structures widely used in various other areas where manipulation and computation with Boolean functions is required. BDD is a representation of a Boolean expression using a rooted directed acyclic graph that consists of terminal nodes (with constant values 0 or 1) and non-terminal nodes (marked with variables). A reduced ordered binary decision diagram, which is a widely used data structure in practice, is a BDD with a particular variable order where redundant nodes are shared, and redundant subtrees are also shared. ROBDDs are derived by the reduction of the corresponding binary decision tree (BDT). ROBDDs provide a compact representation allowing one to process large Boolean functions efficiently in terms of space and time [7].

The most important characteristic of the ROBDD is the size of the graph representation or number of nonterminal nodes. This parameter is critical since the memory requirement during the construction of an ROBDD is directly proportional to the size [2]. Besides the size, the following basic parameters are most often considered: the number of paths, width, and the average path length. There is a direct correspondence between these characteristics and the basic characteristics of a logic network derived from ROBDDs. For example, the size of an ROBDD corresponds to the number of elementary modules in the corresponding realization of a logic network [1]. The paths related characteristics directly correspond to the interconnection complexity of the logic realization.

Therefore, this paper proposes researching the basic ROBDD characteristics of bent functions. Previously, Shafer [9], has analyzed the ROBDD characteristics of disjoint quadratic bent functions, symmetric bent functions, and homogeneous bent functions of 6-variables. Specifically, disjoint quadratic bent functions were found to have size $2n - 2$ for

functions of *n*-variables, symmetric bent functions have size $4n - 8$, and all homogeneous bent functions of 6-variables were shown to be P-equivalent. Two functions are P-equivalent iff those two functions have identical BDDs for distinct variable orderings [9]. However, in this paper the complete set of bent functions with 4 variables is analyzed. For bent Boolean functions with 6 or 8 variables, only appropriate subsets of bent functions are analyzed and not only the size is included, but also the number of paths, width, and the average path length.

A decision diagram experimental framework has been used for implementation of a program for the calculation of ROBDD characteristics. For discovery of bent functions, it uses maximal algebraic degree as the search space boundary. Also, it uses the implementation of the discovery of bent functions using Reed-Muller (RM) subsets, which is described in [10]. Experimental results show interesting ROBDD characteristics of bent functions. For each ROBDD characteristic, the range of values is determined. Additionally, this paper also investigates the same ROBDD characteristics of non-bent functions with *n*-variables having Hamming weight equal to $(2^{n-1} \pm 2^{(n/2)-1})$. These additional characteristics confirm that experimental results can be used to create methods for discovering bent functions using ROBDD.

This paper is organized as follows: Section 2 shortly introduces the theoretical background about bent functions and their discovery in the RM domain. Section 3 discusses ROBDDs and their characteristics. The experimental results are shown and discussed in Section 4. The closing Section 5 summarizes the results of the research reported in this paper.

2. BACKGROUND THEORY

If a function f and its Walsh spectrum $S_{f,w}$, in matrix notation, are represented by vectors $F = [f_{0,}, f_1, \ldots, f_{2^n-1}]^T$, and $S_{f,w} = [s_0, s_1, , \ldots, s_{2^n-1}]^T$ respectively, the Walsh transform is defined by the Walsh matrix $W(n)$ [11]:

$$S_{f,w} = W(n)(-1)^F \tag{1}$$

$$f(x_1, \ldots, x_n) = 2^{-n} X(n) S_{f,w} \tag{2}$$

where

$$X(n) = \bigotimes_{i=1}^{n} \begin{bmatrix} 1 & 1 - 2x_i \end{bmatrix}, \quad (-1)^F = [(-1)^{f_0}, (-1)^{f_1}, \ldots, (-1)^{f_{2^n-1}}]^T \tag{3}$$

and

$$W(n) = \bigotimes_{i=1}^{n} W(1), \; W(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4}$$

where $W(1)$ represents the basic Walsh transform matrix and $\otimes$ is the Kronecker product.

The nonlinearity of an *n*-variable Boolean function f is the (Hamming) distance of f from the set of all *n*-variable affine functions [3]. Boolean functions achieving maximal nonlinearity are called bent functions. Every bent function has a Hamming weight (number of times it takes the value 1) of $2^{n-1} \pm 2^{(n/2)-1}$. For example, bent functions with 4 variables have Hamming weight of 6 or 10, with 6 variables have 28 or 36, and with 8 variables have 120 or 136.

A Boolean function f in (1,–1) encoding is bent if all Walsh spectral coefficients $S_{f,w}$ have the same absolute value $2^{n/2}$.

The fast transform algorithm can be used to compute the coefficients in Walsh spectrum. This algorithm is composed of the "butterfly" operations which are repeated and have structure derived from the basic transform matrices [11]. The recursive definition of the Walsh transform matrix, expressed in Eq. (4), is the fundamental for the definition of the fast Walsh transform algorithm similar to a fast Fourier transform (FFT) algorithm. The computation of the fast transform algorithm consists of the repeated application of the same "butterfly" operations determined by the basic transform matrices [10]. Figure 1 shows the "butterfly" operation for the Walsh transform matrix. The "butterfly" operation are performed in each step over a different subset of data. Figure 1 also shows the flow graphs of the fast Walsh transform algorithm of the Cooley-Tukey type for computation of the Walsh spectrum of a 2-variable Boolean function f given by the truth-vector $F = [f(0), f(1), f(2), f(3)]^T$. This algorithm is highly exploited for testing of bentness across all possible Boolean functions in some defined space for their discovery.

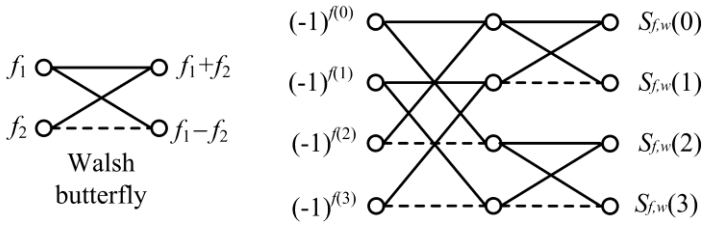

Fig. 1 The flow graph of the fast Walsh transform for 2-variable Boolean function

For example, bentness testing for a function of 4 variables $f(x_1,x_2,x_3,x_4)$, given by truth vector $F = [1,0,1,1,0,1,0,0,0,1,0,0,0,1,0,0]^T$, with (1,–1) encoding, can be calculated as shown on Figure 2. It should be noticed that the absolute values of all Walsh coefficients are equal to 4.

A positive polarity Reed-Muller form comprises exclusive-OR of AND product terms, where each variable appears uncomplemented. Any Boolean function f can be represented by the positive polarity RM form in matrix notation defined as [11]:

$$S_{f,RM} = R(n)F \tag{5}$$

$$f(x_1,\ldots,x_n) = X(n)S_{f,RM} \tag{6}$$

where

$$X(n) = \bigotimes_{i=1}^{n}\begin{bmatrix} 1 & x_i \end{bmatrix} \tag{7}$$

and

$$R(n) = \bigotimes_{i=1}^{n} R(1),\ R(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},\ R(1) = (R(1))^{-1}, R(n) = (R(n))^{-1} \tag{8}$$

where addition and multiplication are modulo 2, $R(n)$ is the positive Reed-Muller transform matrix of order n, and $R(1)$ is the basic positive Reed-Muller transform matrix.

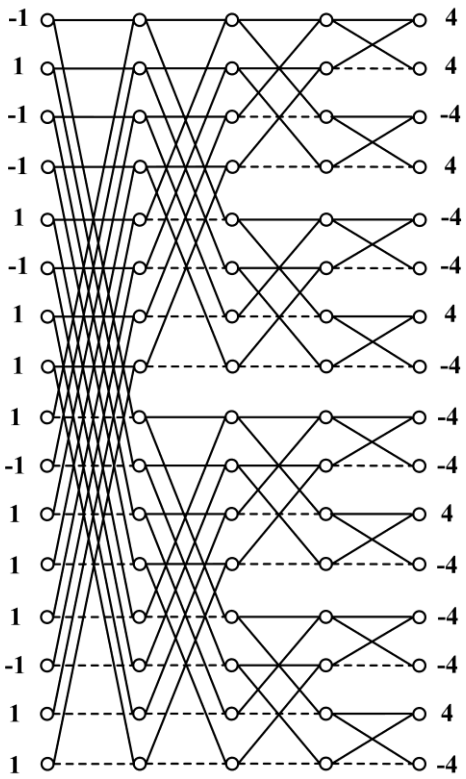

Fig. 2 Example of the bentness testing for a 4-variable Boolean function in the (1, -1) encoding using the fast Walsh transform

The elements of $S_{f,RM} = [a_{0,}a_1, a_2, a_{12}, a_3, a_{13}, a_{23}.a_{123},..., a_{12...n}]$ are coefficients in the positive polarity Reed-Muller (PPRM) espressions for any Boolean function [11]:

$$f(x) = a_0 \oplus \sum_{i=1}^{n} a_i x_i \oplus \sum_{1 \le i < j \le n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n \tag{9}$$

where Σ denotes modulo 2 summation.

The algebraic degree or the order of nonlinearity of a Boolean function f is a maximum number of variables in a product term with non-zero coefficient a_k, where k is a subset of {1,2,3,...,n}. When k is an empty set, the coefficient is denoted as a_0 and is called the zero-order coefficient. Coefficients of order 1 are $a_1, a_2, \ldots a_n$, coefficients of order 2 are $a_{12}, a_{13}, \ldots a_{(n-1)n}$, coefficient of order n is $a_{12\ldots n}$. The number of all coefficients of order i is $\binom{n}{i}$. The PPRM coefficients are divided into order groupings according to the number of ones in the binary representation of its index in the spectrum.

The algebraic degree of bent functions is at most $n/2$ for $n \geq 4$ [8]. Thus, the maximal number of non-zero PPRM coefficents of a bent functions is: $\sum_{i=0}^{n/2} \binom{n}{i}$.

Since the order of Bent functions is limited, the number of non-zero PPRM coefficents is also limited and the positions of the coefficients in the PPRM spectrum are restricted. These restrictions are the main reasons for discovery possibility since they certainly reduce the possible search space for discovery in the Reed-Muller domain. As the Boolean function size increases, the possible search space increases too.

For the PPRM transform, we need an inverse transform to get back from the Reed-Muller domain. Since the Reed-Muller transform matrix $R(n)$ is a self-inverse matrix over $GF(2)$, the forward and inverse transform are given by the same matrix.

Figure 3 shows the flow graphs of the fast inverse Read-Muller transform algorithm of the Cooley-Tukey type for computation of the Boolean function f with the truth- vector $F = [f(0), f(1), f(2), f(3)]^T$ from a PPRM spectrum. This algorithm is highly exploited for discovery of bent functions across all possible Boolean functions in the RM domain.

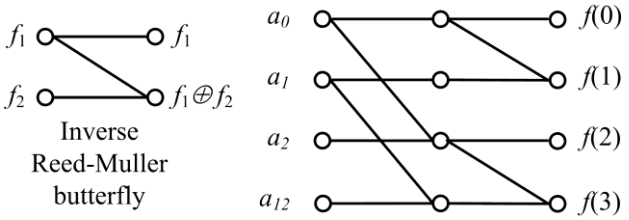

Fig. 3 The flow graph of the fast inverse RM transform for 2-variable Boolean function

For example, discovery of the bent function for a function of 4 variables $f(x_1,x_2,x_3,x_4)$, with truth vector $F = [1,0,1,1,0,1,0,0,0,1,0,0,0,1,0,0]^T$, using the fast inverse RM transform of its $S_{f,RM} = [1,1,0,1,1,0,0,0,1,0,0,0,1,0,0,0]$, is shown in Figure 4. The bentness testing for this truth vector, with (1,−1) encoding, is performed as shown in Figure 2. Black dots on the flow graph on the left side in Figure 4 indicate 11 possible positions for non-zero PPRM coefficients. The number of these possible positions is calculated according to the following formula:

$$\sum_{i=0}^{4/2} \binom{4}{i} = \binom{4}{0} + \binom{4}{1} + \binom{4}{2} = 1 + 4 + 6 = 11 \tag{10}$$

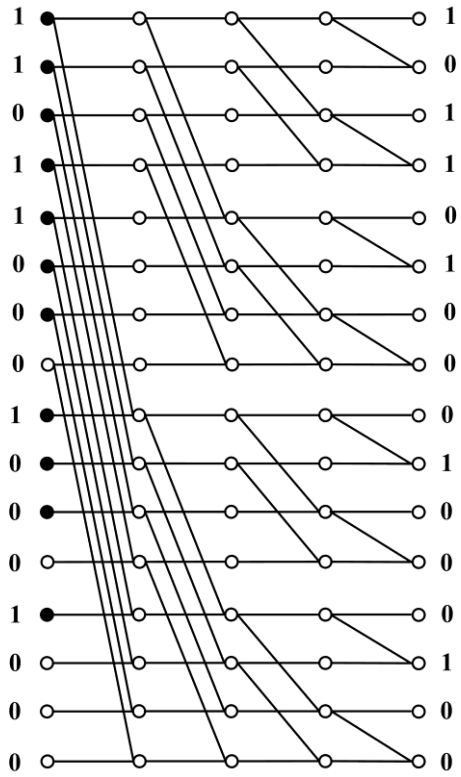

Fig. 4 Example of the discovery of the bent function of 4 variables, using the fast inverse RM transform

This means that there is one possible position for non-zero coefficient of the 0-th order, 4 possible positions for non-zero coefficients of the 1-st order and 6 possible positions for non-zero coefficients of the 2-nd order. The PPRM espression for the function from Figure 4 is:

$$f(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_2 \oplus x_4 \oplus x_1 x_2 \oplus x_3 x_4 \tag{11}$$

Fast transform algorithms are highly exploited for discovering bent functions in the RM domain.

3. ROBDD

A BDD is a directed acyclic graph that contains non-terminal nodes, two terminal nodes, and edges. An ROBDD is a reduced BDD for which the nodes at a same level are labelled with the same variable [12]. The reduction is performed by sharing the isomorphic subtrees and removing the redundant data in the BDT using the appropriately defined reduction rules [6]. Non-terminal nodes are labeled with variables x_i and have two outgoing edges. Outgoing edges are labeled ‘0’ and ‘1’ according to the values of the variable x_i. Terminal nodes contain the function values ‘0’ and ‘1’. The truth table entry

of a Boolean function labels edges from the root node to the corresponding terminal node. An example of the ROBDD representation for the function defined by the truth vector $F=[0,1,1,1,1,0,0,0,1,0,0,0,1,0,0,0]^T$ using ordering (x_1,x_2,x_3,x_4) is shown in Fig. 5.

For characteristics of the ROBDD, the following basic parameters are most often considered: the size, the number of paths, width, and the average path length (APL) [13]. The efficiency of the ROBDD representation in the above example is that it represents a truth vector with a high level of redundancy in a compact form using non-terminal nodes, as long as the data is encoded in such a way that the redundancy is exposed. In a ROBDD for logic function f, the size of the ROBDD is the number of non-terminal nodes needed to represent the ROBDD. In the memory representation of the ROBDD, each non-terminal node requires an index and two pointers to the succeeding nodes.

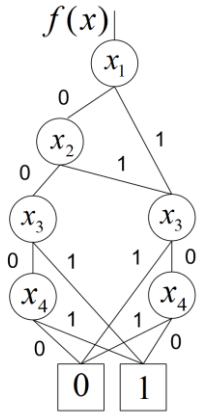

Fig. 5 The ROBDD representation of the function defined by F=[0111 1000 1000 1000]^T

The width of a ROBDD is defined as the maximum number of nodes per level. The delay in a logic network is directly proportional to the number of levels of the ROBDD, which together with the width determines the surface area of the logic network [7].

A path in an ROBDD is the sequence of nodes connected by edges leading from the root node to the terminal node. The number of paths is the sum of all different paths to any of the terminals. The number of paths influences the ROBDD complexity. A minimized disjoint-sum-of-product representation can directly be extracted from an ROBDD and leads to a small logic network [8].

Path length is the number of non-terminal nodes on the path. APL represents the arithmetic mean of the lengths of all possible BDD paths [13]. The minimization of the APL leads to reduction of the logic network evaluation time.

An example of the characteristics of ROBDD representation shown in Fig. 5 is given by:

$$\begin{aligned} &size(ROBDD(f)) = 6 \\ &\#\,paths(ROBDD(f)) = 9 \\ &width(ROBDD(f)) = 2 \\ &APL(ROBDD(f) = 3.66666 \end{aligned} \tag{12}$$

4. Experimental Results

This section presents the basic ROBDD characteristics of bent functions with the initial order of variables as shown in Fig. 5. In these experiments, the order of the variables in the ROBDD was not changed. The complete set of all bent functions is analyzed only for functions of 4 variables. Due to the very time-consuming process for finding bent functions of 6 and 8 variables, the complete set of all bent functions is not analyzed for these cases. For bent functions of 6 and 8 variables, ROBDD characteristics were analyzed on a set of 1 million, and 10,000 bent functions, respectively. The following ROBDD characteristics are included in the analysis: the size (the number of nodes), the number of paths, width, and the average path length. For discovery of bent functions of 4 variables, the maximal algebraic degree is used as the search space boundary. Also, the implementation of the discovery of bent functions of 6 and 8 variables uses RM subsets described in [9]. This implementation performs discovery of single random bent function. Additionally, these experiments also show the number of non-bent functions with n-variables having Hamming weight equal to $(2^{n-1}\pm2^{(n/2)-1})$. Only functions that have a predefined Hamming weight are presented because they represent the search space when creating a potential method for discovering bent functions using ROBDD characteristics.

Implementation of the program for analysis of bent and specific non-bent functions was created using a decision diagram experimental framework. Implementation is done by extension of an existing BDD package using the C++ programming language. The BDD package is implemented using all basic recommendations for programming BDD packages (unique table, operation table, garbage collector, swapping levels, etc.) [14], [15].

The experiments are performed on a PC Pentium IV running at 3.66 GHz with 8 GB of RAM. The BDD package performs operations using shared BDDs, but in these experiments, only the shared BDDs with one output were used. The size of the unique table and the operation table was limited to 262,139 entries. Garbage collection was activated when available memory ran low. Tables in this section present for each ROBDD characteristic (parameter) the total number of bent and specific non-bent functions that have that characteristic.

Table 1 shows the number of bent functions of 4 variables with respect to the ROBDD size, number of paths, widths, and APL. It can be noticed that almost 70% of all bent functions of 4 variables have size 7 or 8, about 63% of functions have the number of paths 9, 10 or 11, about 58% of all bent functions of 4 variables have the ROBDD width 2, and about 40% have the ROBDD APL 3.33333 or 3.5.

This table also shows the number of all non-bent functions having Hamming weight equal to 6 or 10 of 4 variables with respect to the ROBDD size, width, number of paths, and APL. It is evident that it is necessary to average 20 checks of functions that have Hamming weight 6 or 10 that one of them to be bent.

Table 2 shows the number of bent functions of 6 variables on a sample of 1 million functions with respect to the ROBDD size, number of paths, width, and APL. The reason why the entire set of bent functions with 6 variables was not tested is the long time it took to discover these functions. In this table, about 70% of the sampled bent functions of 6 variables have size 15,16, or 17, about 50% have the number of paths 22, 23,24 or 25, about 73% have a ROBDD width 2, and about 11% have the ROBDD APL 5.00007 or 5.125.

Similarly, this table also shows the number of non-bent functions of 6 variables having Hamming weight equal to 28 or 36 on a sample when there are 1,000,000 discovered bent

functions with respect to the ROBDD size, number of paths, width, and APL. It can be noticed that the number of non-bent functions follows the number of bent functions with a ratio of about 40 times more.

Table 3 shows the number of bent functions of 8 variables on a sample of 10,000 functions with respect to the ROBDD size, number of paths, width, and APL. The reason why the number of tested functions is reduced to 10,000 is the very long computation time required for discovery of these functions. In this table, about 60% of the sampled bent functions of 8 variables have size 22, 23, or 24, about 70% have the number of paths 30, 31, or 32. About 90% of the sampled bent functions have ROBDD width 2, and about 50% have ROBDD average path lengths 6.12903, or 6.133335.

This table also shows the number of non-bent functions of 8 variables having Hamming weight equal to 120 or 136 on a sample when there are 10,000 discovered bent functions. It can be noticed that the number of non-bent functions follows the number of bent functions with a ratio of about 80 times more.

If we look at all three tables for functions with 4, 6 and 8 variables, it is easy to determine the formulas for the characteristics that have the largest number of functions. It is discovered that the largest number of n-variable bent functions have size $4*n-8$. Regarding other ROBDD characteristics for the maximum number of functions, no law can be determined. It is interesting that there is a law for bent functions of 4 and 6 variables where APL has formula $0.8333333*n$.

Table 1 The number of all bent and all non-bent functions of 4 variables having Hamming weight equal to 6 or 10 with respect to the ROBDD parameters.

size	#*f* (bent)	#*f* (non-bent)
4	7	169
5	44	667
6	153	2001
7	308	4079
8	318	5296
9	66	2630

width	#*f* (bent)	#*f* (non-bent)
1	132	1780
2	520	8531
3	216	4113
4	28	442

#paths	#*f* (bent)	#*f* (non-bent)
5	3	90
6	17	272
7	52	728
8	117	1471
9	193	2398
10	210	2985
11	157	3074
12	95	2294
13	42	1186
14	9	323
15	1	33

APL	#*f* (bent)	#*f* (non-bent)
2.4	3	61
2.66667	12	123
2.83333	5	92
2.85714	12	176
3	38	572
3.125	78	907
3.14286	2	30
3.22222	8	430
3.25	39	484
3.33333	172	1740
3.4	42	1468
3.44444	13	227
3.5	168	1529
3.54545	86	2405
3.63636	71	669
3.66667	86	2143
3.75	9	152
3.76923	42	1182
3.85714	9	323

Table 2 The number of bent and non-bent functions of 6 variables having Hamming weight equal to 28 or 36 with respect to the ROBDD parameters.

size	#*f* (bent)	#*f* (non-bent)
8	3	142
9	49	2168
10	369	15845
11	2078	84473
12	9182	271672
13	33303	1005621
14	94595	2559252
15	195267	7037689
16	273064	10165891
17	225355	8587546
18	121807	5018437
19	37694	1889635
20	6794	506954
21	440	35829

width	#*f* (bent)	#*f*(non-bent)
2	269858	8434452
3	590964	22425672
4	133894	4584176
5	5284	155319

#paths	#*f* (bent)	#*f* (non-bent)
11	3	194
12	26	1593
13	120	5926
14	397	18922
15	1586	68573
16	3643	151345
17	7840	302133
18	20466	772155
19	37087	1363662
20	61213	2197944
21	83862	3018102
22	116633	4201566
23	135603	5127539
24	136380	6056333
25	117712	5578164
26	97858	4783127
27	80974	4205942
28	46092	2589411
29	26284	1881356
30	16647	1204724
31	8095	712410
32	1467	166751
33	12	1478

APL	#*f* (b)	#*f* (n-bent)
3.90909	3	122
3.91667	12	568
4.07692	24	799
4.08333	6	188
4.15385	32	1420
4.16667	8	521
4.21429	48	2278
4.23077	12	499
4.25	48	2568
4.26667	284	10891
4.28577	178	6995
4.30786	45	1591
4.3125	70	4575
4.33333	449	20481
4.35714	36	1665
4.38463	7	302
4.4	284	9895
4.41176	200	9678
4.42857	72	2889
4.4375	1254	59642
4.46667	282	9612
4.47059	96	4576
4.50019	823	47612
4.52941	660	28902

APL	#*f* (bent)	#*f* (n-bent)
4.80952	868	44599
4.8125	13	568
4.81818	4922	228955
4.82353	103	5601
4.83333	1128	57546
4.84214	4359	178221
4.85	5518	316647
4.85729	9472	376566
4.86431	12929	564982
4.88235	234	9677
4.89596	2046	74766
4.90063	34213	1956687
4.90478	15467	602886
4.90909	2191	80885
4.91304	35656	2012624
4.91667	18849	604202
4.9447	147	6457
4.95023	2078	74702
4.95238	25216	870164
4.95424	398	19465
4.95456	45164	1972354
4.95652	11110	575503
4.95833	639	26774
5.00007	68074	3089332

APL	#*f* (bent)	#*f* (n-bent)
5.16667	6967	257702
5.17241	2640	98321
5.17391	2862	97023
5.17857	20647	860785
5.18182	84	3677
5.18518	14512	798544
5.19231	757	34577
5.2	18960	870556
5.20833	13085	561017
5.21429	2091	79045
5.21739	24	885
5.22222	8080	387122
5.23077	22424	874322
5.23333	1849	70554
5.2381	9	349
5.24	4806	156855
5.24138	11841	590446
5.25	9424	519445
5.25806	5351	254023
5.25926	1928	85266
5.26087	38	1671
5.26667	9674	385661
5.26923	5140	202677

4.53333	14	569
4.55556	656	30187
4.56383	900	35671
4.57143	2	85
4.58916	3783	165761
4.60025	221	8647
4.61111	106	4854
4.62533	220	12587
4.63158	1266	52908
4.64706	1362	49762
4.65	6022	158712
4.66667	10588	458972
4.68421	7271	276134
4.6875	36	1589
4.7	1436	57125
4.7059	1276	45476
4.72228	6569	245564
4.73684	1609	47933
4.75	2227	78231
4.7619	7652	256443
4.76471	90	3554
4.77778	1040	39677
4.78947	19927	819556
4.80001	8330	122

5.04167	5929	258466
5.04348	42467	1704665
5.04545	29139	956027
5.04762	172	5702
5.05	52	1795
5.0527	96	3899
5.07692	13845	507879
5.08	36418	1422644
5.08333	10918	403354
5.08696	6941	353002
5.09117	13059	680223
5.09532	574	31066
5.10649	8	346
5.10714	5499	285302
5.11111	39390	1896601
5.11538	23450	873321
5.12	3655	154667
5.125	45599	1564998
5.13043	31476	1502337
5.13636	234	13121
5.14815	3456	165209
5.15067	39	1466
5.15385	19933	698680
5.16	32140	1570665

5.27586	2173	75331
5.28	4278	165006
5.28571	2793	95661
5.29167	24	1164
5.2963	12014	485664
5.30769	2616	106447
5.31034	3726	185433
5.31818	1	43
5.32143	3222	135886

Table 3 The number of bent and non-bent functions of 8 variables having Hamming weight equal to 120 or 136 with respect to the ROBDD parameters

size	*#f (bent)*	*#f (non-bent)*
16	9	1040
17	37	4074
18	110	12061
19	196	19866
20	438	41860
21	960	91649
22	1598	137048
23	2006	160217
24	2518	219748
25	1340	121521
26	740	67832
27	40	3821
28	8	836

width	#f (bent)	#f (non-bent)
2	9110	712085
3	890	83305

APL	*#f (bent)*	*#f (non-bent)*
5.80769	300	40801
5.83333	30	2161
5.84	120	9212
5.91304	15	1408
5.91667	60	6304
5.92593	88	6602
5.96	16	1282
5.96154	44	3240
6	24	1960
6.0303	540	52486
6.03333	48	3595
6.03448	36	3342
6.03571	114	9354
6.03704	16	1750
6.03846	8	705
6.04	22	1886
6.04167	8	711
6.0625	216	17381
6.06452	54	3865
6.07143	24	1938
6.07407	44	3982

APL	*#f (bent)*	*#f (non-bent)*
6.11765	36	3296
6.12	4	341
6.12903	3600	269403
6.13333	1329	124259
6.14286	136	9576
6.14815	38	2902
6.15152	36	2920
6.15385	22	2114
6.15625	18	1390
6.16	5	84
6.16667	72	5751
6.17241	402	28489
6.18182	48	4112
6.2	40	2971
6.21875	492	40242
6.22222	68	5509
6.22581	233	22063
6.23077	13	1071
6.24138	64	5356
6.25	215	23555
6.25806	12	860

#paths	#f (bent)	#f (non-bent)
23	15	1661
24	98	9720
25	167	16465
26	405	34536
27	261	21638
28	527	41872
29	925	73051
30	1597	124734
31	3899	331211
32	1446	125449
33	624	63760
34	36	4603

6.07692	10	764
6.09375	720	52980
6.10345	372	28586
6.11538	8	706

6.26667	102	7354
6.32143	14	1166
6.33333	13	1385
6.34483	51	4902

5. Conclusions and Future Work

One efficient way to represent Boolean functions is with a reduced ordered binary decision diagram. The strength of ROBDDs is that they can represent Boolean function data with a high level of redundancy in a compact form. The quality of compactness is expressed by basic ROBDD parameters or characteristics. These basic characteristics are the size, the number of paths, the width, and the average path length. Using these characteristics, bent function analysis can be performed to determine their properties better. This paper investigates the characteristics of bent functions with a focus on their basic ROBDD parameters. A decision diagram experimental framework has been used for implementation of a program for calculation of these parameters. The complete set of all bent functions is analyzed for functions of 4 variables. Due to very time-consuming process for the discovery of bent functions of 6 and 8 variables, ROBDD characteristics were analyzed on a set of 1 million, and 10,000 bent functions, respectively. So that we can use these experimental results in future research, this paper also investigates the ROBDD characteristics of non-bent functions with n variables having Hamming weight equal to $(2^{n-1}\pm 2^{(n/2)-1})$ with focus on the same parameters. The complete set of all non-bent functions of 4 variables is analyzed. The set of non-bent functions of 6 variables is analyzed on a sample when there are 1 million discovered bent functions and the set of non-bent functions of 8 variables is analyzed on a sample when there are 10,000 discovered bent functions.

From the experimental results, it is evident that for bent functions of 4 variables there is a small set of values of ROBDD characteristics that most bent functions have. For example, for these functions, 70% of them have the ROBDD size of 7 or 8, 63% have the number of paths of 9, 10 or 11. 58% have the width 2 and 40% have the average path lengths 3.33333 or 3.5. For bent functions of 6 variables, the values of the ROBDD characteristics that have the largest number of bent functions can be determined again. The same applies to bent functions with 8 variables. It was also determined that the largest number of n-variable bent functions has a size of $4*n-8$, and an average path length of bent functions of 4 and 6 variables is very close to $0.8333333*n$. But unfortunately, it was not possible to confirm the same average path length formula for bent functions of 8 variables. Perhaps the reason for this is the small set of functions that was tested.

From the experimental results for non-bent functions, it is evident that they follow the characteristics of bent functions. The ratio of the number of non-bent to bent is about 20 times more for functions of 4 variables, about 40 times more for functions of 6 variables and about 80 times more for functions of 8 variables. These values also represent the search space when creating a potential method for discovering bent functions using ROBDD characteristics.

The results presented in this paper are intended to be used to create methods for the construction of bent functions using ROBDD as a data structure from which the bent functions can be discovered. Research in this direction can reduce the time for discovering random bent functions. In addition, the results in this work represent new boundaries within which we can detect bent functions. It was shown that a large percentage of bent functions with 4, 6 and 8 variables have a very small range of ROBDD characteristics which are tested in this paper. Also, based on individual ROBDD characteristics, new subsets of bent functions can be defined. Bent function discovery can be performed within these ROBDD subsets that have a predefined Hamming weight.

Future work will refer to the study of a pair or more ROBDD parameters of bent functions. It also can be extended to research of additional ROBDD parameters of bent functions, as well as to the study of the characteristics of not only binary decision diagrams, but also other types of diagrams, such as functional decision diagrams, algebraic decision diagrams, Kronecker decision diagrams, Pseudo-Kronecker decision diagrams, etc [7], [8], [11].

REFERENCES

[1] O. Rothaus, "On Bent Functions", *J. Comb. Theory Ser. A*, vol. 20, pp. 300-305, 1976.
[2] O. Logachev, A. Salnikov and V Yashchenko, *Boolean Functions in Coding Theory and Cryptography*, American Mathematical Society, 2012.
[3] S. Mesnager, *Bent Functions, Fundamentals and Results*, Springer International Publishing, 2016.
[4] N. Tokareva, *Bent Functions, Results and Applications to Cryptography*, Academic Press, 2015.
[5] M. Stanković, C. Moraga and R. Stanković, "An improved spectral classification of Boolean functions based on an extended set of invariant operations", *FU: Elect. Energ.*, vol. 31, no. 2, pp. 189-205, 2018.
[6] P. Langevin and G. Leander, "Counting all bent functions in dimension eight 99270589265934370305785861242880", in *Designs, Codes and Cryptography*, vol. 59, pp. 193-201, 2011.
[7] T. Sasao and M. Fujita, *Representations of Discrete Functions*, Kluwer Academic Publishers, Boston, 1996.
[8] R. Drechsler and B. Becker, *Binary Decision Diagrams: Theory and Implementation*, Springer US, 2013.
[9] N. Schafer, "The Characteristics of the Binary Decision Diagrams of Bent Functions", M.S. Thesis, Naval Postgraduate School, Monterey, CA, September 2009.
[10] M. Radmanović, "Efficient Discovery of Bent Function Using Reed-Muller Subsets", In. Proceedings of the 55th Int. Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST 2020), pp. 7-10, 2020.
[11] M. G. Karpovsky, R. S. Stanković and J. T. Astola, *Spectral Logic and Its Applications for the Design of Digital Devices*, Wiley, 2008.
[12] M. Thornton, R. Drechsler and D. Miller, *Spectral Techniques in VLSI CAD*, Springer US, 2012.
[13] S. Nagayama, A. Mishchenko, T. Sasao and J. T. Butler, "Minimization of average path length in BDDs by variable reordering", In Proceedings of the International Workshop on Logic and Synthesis, 2003, pp. 207-213.
[14] K. Brace, R. Rudell and R. Bryant, "Efficient implementation of a BDD package", In Proceedings of the 27th ACM/IEEE Design Automation Conference, 1990, pp. 40-45.
[15] F. Somenzi, "Efficient manipulation of decision diagrams", *Software Tools for Technology Transfer*, vol. 3, no. 2, pp. 171-181, 2001.